

**We Claim:**

1. A method for generating a signal rich in prosody information comprising:  
a first step including in said signal a plurality of phoneme symbols,  
a second step including in said signal a desired duration of each of said phoneme  
symbols,  
a third step including at least one target prosody parameter value within a duration  
for at least one of said phonemes at a time offset from the beginning of the duration of said  
phoneme that is greater than zero and less than the duration of said phoneme.

5 2. The method of claim 1 where said prosody parameter is pitch.

3. The method of claim 1 where said prosody parameter is energy.

15 4. The method of claim 1 where said third step includes target values for both pitch  
and energy.

20 5. The method of claim 1 where at least some of the phonemes have no prosody  
parameter targets specified for the beginning of the durations of said at least some of the  
phonemes.

6. The method of claim 1 where timing of said prosody parameter target  
specifications are expressed in terms of durations.

7. The method of claim 1 where timing of said prosody parameter target  
specifications are expressed in terms of time offsets from the beginning of durations of  
phonemes.

8. The method of claim 1 where at least some silence intervals have one or more  
prosody parameter target specifications.

30 9. The method of claim 1 where the format of said signal is:

```

Dur_Enable
F0_Contour_Enable
Energy_Contour_Enable
Number_of_Phonemes
5 Phonemes_Symbols_length
for (j=0;j<Phoneme_Symbols_Length; j++)
    Phoneme_Symbols
for (j=0; j<Number_of_Phonemes; j++) {
    if(Dur_Enable)
        Dur_each_Phoneme
10     if (F0_Contour_Enable) {
        num_F0
        for (j=0; ,<num_FO; j++) {
            F0_Countour_Each_Phoneme
            F0_Countour_Each_Phoneme_time
            }
        }
        if (Energy_Contour_Enable)
            Energy_Countour_Each_Phoneme
20
}

```

*Sub A3* 10. The method of claim 9 where said signal also includes text specifications.

25 11. The method of claim 1 where the format of said signal is:

Dur_Enable	1
F0_Contour_Enable	1
Energy_Contour_Enable	1
Number_of_Phonemes	10
30 Phonemes_Symbols_length	13
for (j=0;j<Phoneme_Symbols_Length; j++)	

	Phoneme_Symbols	8
	for (j=0; j<Number_of_Phonemes; j++) {	
	if(Dur_Enable)	
	Dur_each_Phoneme	12
5	if(F0_Contour_Enable) {	
	num_F0	5
	for (j=0; ,<num_F0; j++) {	
	F0_Countour_Each_Phoneme	8
	F0_Countour_Each_Phoneme_time	12
10	}	
	}	
	}	
	if (Energy_Contour_Enable)	
	Energy_Countour_Each_Phoneme	24
15	}	

where the numbers correspond to the number of bits.

**12.** The method of claim 1 the format of said signal is:

	TTS_Sentence_Start_Code	32
20	TTS_Sentence_ID	10
	Silence	1
	if(Silence)	
	Silence_Duration	12
	else {	
25	if(Gender_Enable)	
	Gender	1
	if(Age_Enable)	
	Age	3
	if (!Video_Enable & Speech_Rate_enable)	
30	Speech_Rate	4
	Length_of_Text	12

```

\ For (j=0; j<=Length_of_Text; j++)
    TTS_Text                                8
    if (Prosody_Enable) {
        Dur_Enable                           1
        F0_Contour_Enable                   1
        Energy_Contour_Enable              1
        Number_of_Phonemes                10
        Phonemes_Symbols_length          13
        for (j=0;j<Phoneme_Symbols_Length; j++)
            Phoneme_Symbols               8
            for (j=0; j<Number_of_Phonemes; j++) {
                if(Dur_Enable)
                    Dur_each_Phoneme      12
                    if(F0_Contour_Enable) {
                        num_F0                  5
                        for (j=0; ,<num_F0; j++) {
                            F0_Countour_Each_Phoneme   8
                            F0_Countour_Each_Phoneme_time 12
                            }
                            }
                            if (Energy_Contour_Enable)
                                Energy_Countour_Each_Phoneme 24
                            }
                }
            if (Video_Enable) {
                if (Dur_Enable) {
                    Sentence_Duration       16
                    Postion_in_Sentence      16
                    Offset                     10
                }
            }
        }
    }
}

```

```

        }
      if(Lip_Shape_Enable) {
        Number_of_Lip_Shape
      10
        for(j=0; j<Number_of_Lip_Shape; j++) {
          If(Prosody_Enable) {
            If(Dur_Enable)
              Lip_Shape_Time_in_Sentence
            16
            Else
              Lip_Shape_Phoneme_Number_in_Sentence
            13
          10
          }
        else
          Lip_Shape_Letter_Number_in_Sentence
        12
        Lip_Shape
        8
      15
      }
    }
  
```

where the numbers correspond to the number of bits.

*add  
alt*